Support For
sinclair
ZX81 - spectrum - QL
and
TIMEX sinclair
1000 - 1500 - 2068
computers

TIMELINEZ

VOLUME 5 ISSUE 8 SEPTEMBER 1987 $1.00

THE JOINT NEWSLETTER OF THE THREE TIMEX-SINCLAIR
USER GROUPS IN THE SAN FRANCISCO BAY AREA

EBZUG **PUG** **SVSTUG**

GIF GRAPHICS DECODING-ENCODING NOW AVAILABLE TO SINCLAIR QL USERS

By Norm Lehfeldt

Programmer Don Thompson has once again come to the aid and rescue of Sinclair QL users.

He has released GIF decoding and encoding programs for this machine.

(For those who may not recall, GIF -- Graphics Interchange Format -- is a new protocol established by CompuServe as a means for the exchange of graphics screens among dis-simliar computers. A GIF file begins with a header which describes the size of the image that follows and the colors employed. The decoder must than adapt the image for display on the receiving computer.)

Don's program for accomplishing the complicated decoding task uses sophisticated algorithms that make mathmaticl conversions of the pixel counts of images created on other computers so that their geometry is preserved on the QL screen. In addition the program contains a pallette of 64 stipples which greatly expand the color capabilites of the QL. In many cases, the displayed QL image is nearly indistinguisable for the original picture created on graphically more sophisticated machines.

At present, Don's programs are available to CompuServe subscribers in Data Library 6 of CLUB SIG for only the cost of the connect time necessary to download them.

Don stresses that it is important to read the accompanying DOC file before attempting to use the programs.

-#-

'FONTMAN' VARIES TYPE STYLES ON 2068 SCREEN OR PRINTER

By Norm Lehfeldt

'Font Manager,' Jack Dohaney's latest achievement is a program for creating, editing and printint to screen or printer an endless variety of type styles.

As supplied by Jack, the program includes two libraries of twenty fonts each. But the type design possibilities of the program are unlimited.

Once the program is loaded and running, a particular type font may be 'activated.' Subsequent output to screen or printer will be in the font selected.

The key to the printing operation is a new printer driver using your printer's graphics mode for all printing. A fringe benefit of this scheme is that, for the first time, users of 80-column printers will be able to LPRINT and LLIST the full Timex-Sinclair character set.

Order 'FONTMAN' directly from:
Jack Dohaney
390 Rutherford Ave.
Redwood City, CA 94061

Or call Jack at (415) 367 7781 for further details.

Inside This Issue:

QWIKSORT - by Courtney Du Bois

The purpose of this algorithm is to sort a large array very quickly, using as little memory overhead as possible.

The goal of each pass is to place an element of the array in it's final position. While it does this, it also partitions the array into two subsets: those elements less than or equal to the sort key and those with values greater than the key. Each time the array is sub-divided, the larger is stacked and the other is processed. The process is repeated on each subset until all elements have been processed.

Consider this example:

42 23 74 11 65 58 94 36 99 87

Two index variables I and J with the values 1 and 10 are used. K(I) and K(J) are compared and if no exchange is necessary J is decremented by 1 and the process is repeated. When K(I) >= K(J), they are exchanged. 'I' is incremented by 1 and this processed until another change occurs.

The sequence of exchanges for placing 42 in it's final position where the numbers preceded by an asterisk are being compared is as follows:

```
*42  23  74  11  65  58  94  36  99 *87
*42  23  74  11  65  58  94  36 *99  87
*42  23  74  11  65  58  94 *36  99  87
 36 *23  74  11  65  58  94 *42  99  87
 42  23 *74  11  65  58  94 *42  99  87
 42  23 *42  11  65  58 *94  74  99  87
 42  23 *42  11  65 *58  94  74  99  87
 42  23 *42  11 *65  58  94  74  99  87
 42  23 *42 *11  65  58  94  74  99  87
 42  23  11  42  65  58  94  74  99  87
```

The original array has now been partitioned into two subsets: (36, 23, 11) and (65, 58, 94, 74, 99, 87).

Designed to be used as a subroutine, the program requires an array K and a variable N, where N = the number of elements in the array, as input. It begins by stacking the lower and upper boundaries of the entire array in the arrays L and U. T is the stack pointer. Lines 2040 through 2060 unstacks the boundaries of each unprocessed subset. Lines 3000 through 5090 partitions a subset. K saves the sort key. Lines 6000 through 6100 stacks the boundaries of the larger subset. Lines 10 through 60 are included for demonstration purposes.

This algorithm has two shortcomings:

1. It is not appropriate for small arrays (say 10 elements or less).

2. The worst case for this algorithm is when the array is already sorted.

A worthwhile enhancement to this program would be to check the difference of the subset boundaries at line 2065 and if it is less than or equal to 10, use another sorting method.

NOTE: The 'LN' in line 1004 is the natural logarithmic function. Enter, Function Z (shift and enter then press the Z key) to obtain it. The sort took 4 minutes on the PC and 8.5 minutes on the TIMEX.

Mr. Du Bois is a former member of Timex Sinclair EBZUG, Berkeley, CALIFORNIA.

```
   5 REM  *** QWIKSORT ***
   7 REM BY COURTNEY DU BOIS
   9 FAST
  10 LET N=1000
  15 DIM K(N)
  20 FOR E=1 TO N
  30 LET K(E)=INT (RND*100+1)
  40 NEXT E
  50 GOSUB 1000
  60 STOP
1000 REM INITIALIZE STACK
1004 LET S=INT ((LN N)/(LN 2))
1006 DIM L(S)
1008 DIM U(S)
1009 REM STACK ENTIRE ARRAY
1010 LET T=1
1020 LET L(T)=1
1030 LET U(T)=N
2000 REM SORT
2005 REM STACK EMPTY?
2010 IF T THEN GOTO 2040
2020 PRINT "DONE"
2030 RETURN
2035 REM UNSTACK BOUNDARY INFO
2040 LET L=L(T)
2050 LET U=U(T)
2060 LET T=T-1
2065 REM ANYTHING IN SUBSET?
2070 IF U<=L THEN GOTO 2010
2080 GOSUB 3000
2090 GOTO 2070
3000 REM PARTITION SUBTABLE
3010 LET I=L
3020 LET J=U
3030 LET K=K(I)
4000 REM COMPARE, MOVE LEFT
4010 IF K>=K(J) THEN GOTO 4040
4020 LET J=J-1
4030 GOTO 4010
4035 REM HAVE KEYS CROSSED?
4040 IF J>I THEN GOTO 4070
4045 REM RESTORE KEY
4050 LET K(I)=K
4060 GOTO 6000
4070 LET K(I)=K(J)
4080 LET I=I+1
5000 REM COMPARE, MOVE RIGHT AND
WATCH OUT FOR RIGHT EDGE
5010 IF K<K(I) OR I>=N THEN GOTO
 5040
5020 LET I=I+1
5030 GOTO 5010
5040 IF J<=I THEN GOTO 5080
5045 REM SWAP AND MOVE RIGHT
5050 LET K(J)=K(I)
5060 LET J=J-1
5070 GOTO 4000
5075 REM RESTORE KEY
5080 LET K(J)=K
5090 LET I=J
6000 REM STACK LARGER SUBSET
6010 LET T=T+1
6020 IF I-L<U-I THEN GOTO 6070
6025 REM STACK LEFT SUBSET
6030 LET L(T)=L
6040 LET U(T)=I-1
6050 LET L=I+1
6060 GOTO 6100
6065 REM STACK RIGHT SUBSET
6070 LET L(T)=I+1
6080 LET U(T)=U
6090 LET U=I-1
6100 RETURN
```

FROM THE AUGUST 1987 CATS N/L

TS2068 In's and Out's

BY JAMES JONES

In developing an analog joystick using the ZIP compiler, I found that the routine included in the instructions for reading the joysticks is silly. The same machine code is produced by the compiler with the BASIC functions and is usable by the BASIC interpreter with:

```
3000 OUT 245, 14: LET K=
     IN(246 + player * 256)
```

Notice then to read the left joystick:

```
3000 OUT 245, 14: LET K= IN 502
```

The right joystick:

```
3000 OUT 245, 14: LET K= IN 758
```

And perhaps most importantly both joysticks at the same time:

```
3000 OUT 245, 14: LET K=
     IN 1014
```

That is:

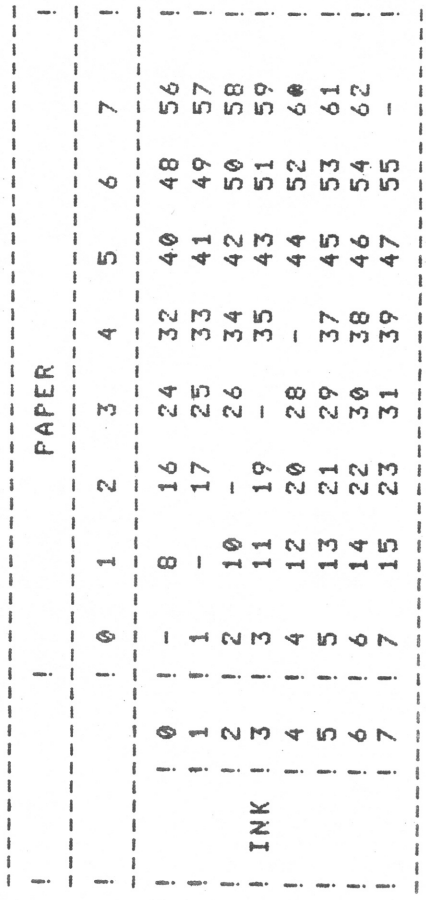
```
3000 OUT 245, 14: LET K=
     IN (246 + 3 * 256)
```

The joysticks are effectively ORed. This could make for some interesting games.

James Jones
2242 Locust
Amarillo, TX 79109
Amarillo TSUG

FROM THE AUGUST 1987 PLOTTER N/L
OF CLACKAMAS COUNTY, OREGON.

COLOR-GRAY SCALE
POKE numbers 1-64 into 23693 with a FOR-NEXT loop to get all colors/grays.
To get BRIGHT add 64 to each number, FLASH add 128, and together add 192.



IN DEFENSE OF PUBLIC DOMAIN SOFTWARE
by PETE FISHER

FROM THE CHICAGO AREA T/S GROUP N/L

-=Shareware/Fairware=-

A fairly new phenomena which has emerged in recent years is the concept of Shareware, also called Fairware or Freeware. In this case, the program is freely exchanged among users, as long as it's accompanying "commercial message" remains intact. This message simply states,"If you liked this program and found it useful, please send $X.00 to the author at:(and gives the address). Many people, of course, never pay. They end up missing out on updates, corrections, bug "fixes" and good quality DOC's. This is the main advantage of Fairware over normal PD. It's a living, ongoing thing with support.

The most famous T/S Fairware author is Jack Dohaney. He calls his enterprise, "Jack's Fairware" and he's currently offering over 20 titles. You can get his latest catalog by sending a S.A.S.E. to Jack at: 390 Rutherford Av./Redwood City, CA 94061. My personal favorites are his MSCRIPT upgrade (which I'm using to write this) and the RLE Decoder/Encoder program called QRL-the best available at this time.

Jack describes Fairware this way,"FAIRWARE is..CONDITIONAL public-domain user-supported software that is NOT exactly free. FAIRWARE costs whatever the USER feels is a fair price for what he or she received.

"The MAIN IDEA behind Fairware is that support is a two- way street: user and author should support each other, for their mutual benefit...."

3-D WIRE FRAME GRAPHICS
By Tim Swenson

The program listed below allows a person to define a 3-D object, draw it on the screen, and rotate it in three different planes.

The heart of the program lies in how the object is represented. Two arrays hold the information for the object. An vertex array (list) holds all of the vertices. An edge array (list) holds the two endpoints of the edges. The edgel array refers back to the vertex list for the actual points. It only holds the number of the vertex.

Lines 100 to 1030 in the program holds the definition of a cube and some values for variables used by the program. Look at the diagrams and the the two lists to see how the cube is defined. More complex figures will take more information.

The center of the cube is (0,0,0). This can be changed by the user. The variables XX, YY, and ZZ define the center that the user chooses.

There are four procedures in the program. They are treated as keywords. To display the object the user can type in directly, or in a program, DISPLAY3D. ROTX, ROTY, and ROTZ rotate the object in radians held by the variable ROT. The program says that ROT is 20, its wrong. It should be in radians, like PI/10. Just to refresh your memory PI/2 is 90 degrees.

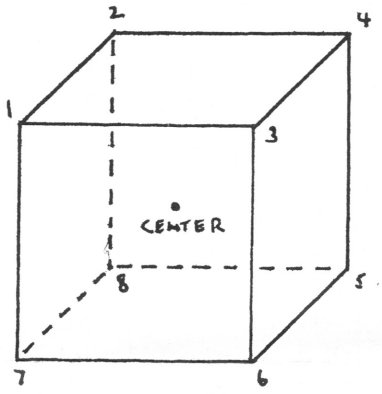
When a rotation procedure is called the object is spun in the axis defined. Since the Z axis is pointing out of the screen, rotating the object will make it appear to spin like it is on a turntable and you are looking down on it. I hope this is clear.

The variable D2 changes how large the object is on the screen. The larger the number the smaller the object.

When you run the program as it is, nothing appears to happen. All it has done is to load the array with the data and set up the variables. Type ROTX, ROTY, and then DISPLAY3D. You will now see the cube at an angle.

Have fun with the program. If you have any questions feel free to contact me. Its hard to summarize a lot of material in a short article.

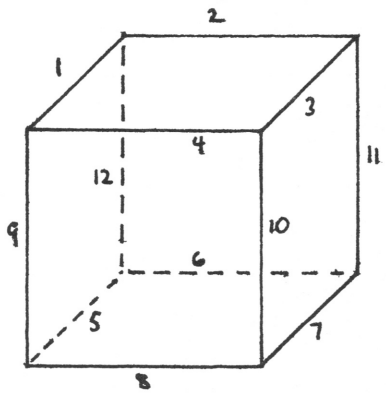
VERTICES



VERTEX LIST

1= (-1,-1,1)	2= (-1,1,1)
3= (1,-1,1)	4= (1,1,1)
5= (1,1,-1)	6= (1,-1,-1)
7= (-1,-1,-1)	8= (-1,1,-1)

EDGES



EDGE LIST

1= (1,2)	2= (2,4)
3= (4,3)	4= (3,1)
5= (7,8)	6= (8,5)
7= (5,6)	8= (6,7)
9= (7,1)	10= (6,3)
11= (5,4)	12= (8,2)

```
100 DIM vertex(100,3)
110 DIM edge(200,2)
190 RESTORE
200 READ xcenter, ycenter
210 READ d1, d2, rot
220 READ xx, yy, zz
230 READ vertexcount
240 FOR loop = 1 TO vertexcount
250  FOR loop2 = 1 TO 3
260    READ vertex(loop,loop2)
270  NEXT loop2
280 NEXT loop
290 READ edgecount
300 FOR loop = 1 TO edgecount
310  READ edge(loop,1)
320  READ edge(loop,2)
330 NEXT loop
1000 DATA 40,40,1,10,20
1010 DATA 0,0,0
1020 DATA 8,-1,-1,1,-1,1,1,1,-1,
1,1,1,1,1,1,-1,1,-1,-1,-1,-1,-1,
-1,1,-1
1030 DATA 12,1,2,2,4,4,3,3,1,7,8
,8,5,5,6,6,7,7,1,6,3,5,4,8,2
9000 DEFine PROCedure rotz
9010  FOR loop = 1 TO vertexcount
9020    x1 = vertex(loop,1)*COS
(rot)-vertex(loop,2)*SIN(rot)
9030    y1 = vertex(loop,1)*SIN
(rot)+vertex(loop,2)*COS(rot)
9040    vertex(loop,1) = x1
9050    vertex(loop,2) = y1
9060  NEXT loop
9090 END DEFine rotz
9100 DEFine PROCedure roty
9110  FOR loop = 1 TO
 vertexcount
9120    x1 = vertex(loop,1)*COS
(rot)-vertex(loop,3)*SIN(rot)
9130    z1 = vertex(loop,1)*SIN
(rot)+vertex(loop,3)*COS(rot)
9140    vertex(loop,1) = x1
9150    vertex(loop,3) = z1
9160  NEXT loop
9190 END DEFine roty
9200 DEFine PROCedure rotx
9210  FOR loop = 1 TO
 vertexcount
9220    y1 = vertex(loop,2)*COS
(rot)-vertex(loop,3)*SIN(rot)
9230    z1 = vertex(loop,2)*SIN
(rot)+vertex(loop,3)*COS(rot)
9240    vertex(loop,2) = y1
9250    vertex(loop,3) = z1
9260  NEXT loop
9290 END DEFine rotx
9300 DEFine FuNction transx
 (x,d1,d2)
9305  LOCal xprime
9310  xprime = (x*d2)/d1
9315  RETurn xprime
9320 END DEFine transx
9330 DEFine FuNction transy
 (y,d1,d2)
9335  LOCal yprime
9340  yprime = (y*d2)/d1
9345  RETurn yprime
9350 END DEFine transy
9400 DEFine PROCedure
 display3d
9405 CLS
9410  FOR loop = 1 TO edgecount
9420  LET point1 = edge(loop,1)
9430  LET point2 = edge(loop,2)
9435  LET x1 = vertex(point1,1)
 - xx
9440  LET x1 = transx(x1,d1,d2)
9445  LET x2 = vertex(point2,1)
 - xx
9450  LET x2 = transx(x2,d1,d2)
9455  LET y1 = vertex(point1,2)
 - yy
9460  LET y1 = transy(y1,d1,d2)
9465  LET y2 = vertex(point2,2)
 - yy
9470  LET y2 = transy(y2,d1,d2)
9480  LINE x1+xcenter,y1+ycenter
 TO x2+xcenter,y2+ycenter
9490  NEXT loop
9495 END DEFine display3d
```

P U G N E W S

by Walt Gaby

I missed the June meeting. But George Mockridge was kind enough to pitch in and report on it.

Since June, there have been two meetings. Here is a brief report on each of them.

* * * * * * * * * * * * *

At the July meeting, Norm Lehfeldt made a plea for help in putting out the newsletter. If you want to make a contribution, please let him know!

Norm also led a discussion on the idea of sponsoring a T/S Computer Faire in the San Francisco Bay Area. Bob Orrfelt is looking into possible facilities.

Jack Dohany indicated that he would like to see a workshop which reviewed programming languages other than BASIC. As examples, he mentioned FORTH, Assembly, Pascal, C, Turtle, and DLAN.

* * * * * * * * * * * * *

At the August meeting, there was more discussion on the proposal for a T/S Computer Faire. Group consensus felt that Silicon Valley would be the best location. Bob gave a summary report on his findings regarding facilities.

Jack Dohany demonstrated his new font program...it is terrific!!! If you are interested, call Jack at (415) 321-7684.

* * * * * * * * * * * * *

Next meeting date: SEPTEMBER 20. See you there!!!

SincLink

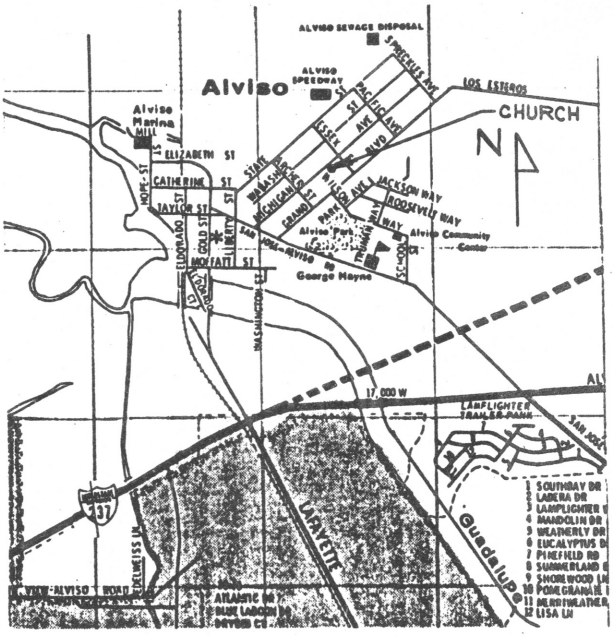
‡ New GUTS/SV Workshop meetings. GUTS/SV is starting a workshop meeting program on Wednesday nights when the regular meeting at 'Star by the Sea' Catholic Church in Alviso is not scheduled. These meetings will be held at members houses for people interested in special projects the member holding the meeting is working on. Check Andy's BBS for the schedule. First meetings will be:
‡‡ Terry Greenlee ... TS Displays
‡‡ Carl Rink ..... A real TS Flight Simulator

‡ New newsletter subscription rates. $15 per year for everyone, to help pay for the new Software library projects which are now available to all subscribers.

‡ GUTS/SV news:

‡‡ 870826 meeting at Alviso 'Star by the Sea' Catholic Church. Meeting Attendees: Oliver Chaplin, Mark & Natalie Wahl, Terry Greenlee, Andy Hradesky, Bill Miller.
‡‡‡ Mark Wahl demoed his 128 column display for the TS2068 on Andy's Monochrome Monitor. It looked pretty readable.
‡‡‡ Andy Hradesky demonstrated the Game 'PHASOR' on the TS1500 and the TS1510 Command Cartridge Adapter (loaned by Bob Orrfelt) and the Command Cartridge program 'Flight Simulator'.
‡‡‡ Bill Miller demonstrated Mark Wahl's new program 'P1', which prints the first sector of each Wafadrive file to the TS2040 printer, the Wafadrive Centronics Port, or a Wafadrive DATA file to easily find out what each program on the Wafer is (in case you can't figure it out from the File Name). It also prints the sectors in Last In First Out (LIFO) order so that the newest files are printed out first. Thanks to Mark for putting this program in the GUTS/SV Library. Maybe he will do a program like this for the other TS Mass Storage Devices?

‡‡ Welcome to new Subscriber Mel Richardson of Windsor, Ontario, Canada.
Mel found out about us from the 'Subscribe to TIMELINEZ/SincLink!' classified ad in TIME DESIGNS MAGAZINE 8705/06 P40.
Thanks to renewing subscribers Dave Means of Fremont, Ca and Jim Wheeler of San Jose, Ca.



September:

19 Sunday 1:00 P.M.

Peninsula Users Group

24 Thursday 7:30 P.M.

East Bay ZX81 Users Group

Call Russ English for location

30 Wednesday 7:00 P.M.

Silicon Valley Group

Star of the Sea Church, Alviso

October:

17 Sunday 1:00 P.M.

Peninsula Users Group

Peninsula Hospital, Burlingame

22 Thursday 7:30 P.M.

East Bay ZX81 Users Group

Call Russ English for location

28 Wednesday 7:00 P.M.

Silicon Valley Group

Star of the Sea Church, Alviso

November:

21 Sunday 1:00 P.M.

Peninsula Users Group

Peninsula Hospital, Burlingame

Important Notice:

Due to Thanksgiving on Thursday November 26, East Bay and Silicon Valley meetings may be cancelled or changed. Call one of the officers for late info.

Is it time to renew your membership?

E B Z U G EAST BAY Z80 USER GROUP
3128 KING STREET
BERKELEY, CALIFORNIA 94703
(Woody McPheeter's)

CONTACT: RUSS ENGLISH (415) 465-3116

MEETINGS: FOURTH THURSDAY OF EACH MONTH, 7:30 P.M.
CALL RUSS ENGLISH FOR DETAILS ON MEETING PLACE.

MAKE CHECKS FOR DUES PAYABLE TO "WOODY McPHEETERS".

P U G PENINSULA USER GROUP
311 MICHELLE LANE
DALY CITY, CALIFORNIA 94015
(415) 878-1773

PRESIDENT: GEORGE MOCKRIDGE

MEETINGS: THIRD SUNDAY OF EACH MONTH, 1:00 P.M.
PENINSULA HOSPITAL
1783 EL CAMINO REAL, BURLINGAME

MAIL DUES TO: PAT MORRISSEY
2000 CRYSTAL SPRINGS ROAD
BUILDING 21, APT. 22
SAN BRUNO, CALIFORNIA 94066

S U S T U G SILICON VALLEY SINCLAIR TECHNOLOGY USER GROUP

6675 CLIFFORD DRIVE
CUPERTINO, CALIFORNIA 95014
(408) 253-3175

PRESIDENT: RITA CARR, (408) 738-2888, X-4579

MEETINGS: LAST WEDNESDAY OF EACH MONTH - 7:00 P.M.
(FOR LOCATION OF MEETINGS SEE ACCOMPANYING SUSTUG NEWS.)

MAKE CHECK FOR DUES PAYABLE TO "SINCLINK".

THIS TIMELINEZ NEWSLETTER IS A JOINT PUBLICATION OF THE THREE TIMEX-SINCLAIR USER GROUPS IN THE S.F. BAY AREA.

NEW MEMBERS AND VISITORS ARE ALWAYS WELCOME!

FOR FULL MEMBERSHIP (INCLUDING PARTICIPATION IN GROUP MEETINGS, THE NEWSLETTER, PROGRAM LIBRARY PRIVILEGES AND SPECIAL EVENTS) SEND $15 ANNUAL DUES TO ONE OF THE ABOVE ADDRESSES WITH THE CHECK PAYABLE AS INDICATED.

READERS OUTSIDE THE SAN FRANCISCO BAY AREA MAY SUBSCRIBE TO THE NEWSLETTER ONLY BY SENDING A CHECK FOR $10 TO ONE OF THE ABOVE GROUPS.

John Ezike has reminded me that there are a number of members in each group who are a little behind in their dues. Fifteen bucks isn't very much to support club activities for a year, so check your status and cough up if it's your turn. Also out of town subscribers are reminded that they need pay only ten bucks a year to continue receiving 'TIMELINEZ.'

George Mockridge has forwarded a clipping from a recent issue of Investor's Daily with the headline 'Sinclair to re-enter British Market.' There are about three sentences describing the Z88 -- the new Sinclair laptop. Pardon me for saying so, but it appears that this machine is lurching on to the market in the same way as the QL -- Months and years of advance hype, repeatedly delayed introductions, promised features omitted from machines finally delivered, etc.

I am curious enough to buy one when they arrive, but I am still asking the question; will Sir Clive be remembered as a man who had ONE really great idea -- the ZX81?

I am sorry to have to report that plans for a T/S Fest in the Bay Area in 1988 appear to have fallen through. Those of us who have been working on the plan have reluctantly concluded that none of us as the time free to devote to the project that it would require.

Bob Orffelt is to be commended on all of his efforts in gathering and presenting options to the members of our ad hoc committee.

Maybe another year.

I have one other unpleasant duty -- it is to report that this is the last issue of 'TIMELINEZ' that I will be editing.

Increasing pressures at work and some personal obligations mean that something has to go and I'm afraid that 'TIMELINEZ' is it. I hope to be able to continue an occasional contribution to the newsletter and I urge you to do the same. It cannot continue unless we all participate.

As of this writing, no new editor has been selected. I am very sorry that I could not have continued as editor for a longer period.

If I may, I'd like to offer a word of advice on behalf of the next editor; everyone who contributes to 'TIMELINEZ' should be aware of the editor's requirements and attempt to accomodate them. The assembly of diversely formatted material is a very time-consuming part of the editor's job. Editors will last longer if we all make their job easier.

NHL

Managing Editor:
Editors:
EBZUG: John Ezike
PUG: George Mockridge
SVSTUG: Bill Miller

Advertising:
PUG:
EBZUG: Bill Strick

N/L Exchange: Rita Carr

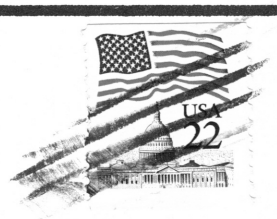
Advertising Schedule:

Full Page:	$40.00
Half Page:	$25.00
1/4 Page:	$15.00
1/8 Page:	$10.00

Published monthly by the Timex/Sinclair User Groups in the San Francisco-Oakland-San Jose areas. Sent to all current members. $15.00 annual dues.

TIMELINEZ
P.O. Box 1312
Pacifica, CA 94044
U.S.A.

First Class Mail



N 08/87 88

Donald S. Lambert
3310 Clover Drive S.W.
Cedar Rapids IA 52404